**4HC3 Assignment 2 Requirements**

* Design and Implement a UI
* Touchscreen-based UI
  + Can assume mouse pointer is fingertip (don’t have to support real touchscreen)
* Can develop in Visual Basic or C#
* Minimal functionality
* GOAL: Design usable interface, support navigation between screens, minimal functionality
* Low grades for: bare minimum functionality, adequate design
* High grades: better design, more functionality
  + More functionality that detracts from usability may not give higher marks
* Design using concepts discussed in class – design principles/guidelines
* Option 1 – Parking Meter:
  + Mechanical Parking Meters are extremely usable
  + Issues:
    - Digital Parking Meter is complicated, error prone
    - Little consistency in design of these
    - Extra instructions taped
    - Spurious buttons with no apparent function or feedback
    - Other major usability issues
  + Requirements:
    - Design digital parking meter UI (touchscreen)
    - Pay a certain amount of cash, leave printed ticket on dashboard indicating how long you can park for
    - Minimum features:
      * Ability to use coins, credit, debit
      * Simulated printing of dash-board ticket (i.e. virtual ticket in dialog box)
      * Ability to add/remove time prior to printing ticket
      * Refund time for returned tickets; for when you paid for more time than you needed. Can returned ticket for partial refund later
* Option 2 – Bank Machine:
  + UI’s are generally simple
  + Users expected to walk up and use with minimal training
  + Issues:
    - Scenario: consider bank machines that accept card, process withdrawal then release card. User is more likely to forget card
    - A swipe could be less error-prone
  + Requirements:
    - Design touchscreen-enabled bank machine UI
    - Minimum features:
      * Ability to enter account number, simulating swiping/inserting a card
      * Ability to withdraw and deposit cash, updating bank account balance accordingly
      * Ability to simply check/display account balances
      * Transfers between accounts
* Other Features:
  + Shouldn’t be tacked on
  + Adding unusable features may reduce grade
  + Well designed minimum functionality > not well designed extra functionality
* Programming Language/Coding Guidelines:
  + Any programming language, no bonus marks for difficult languages
  + Submit in a runnable form (make sure we test this!)
  + Need to provide clear instructions on how to run it
* Submission Notes:
  + Submit code, any project files, and clearly identified executable (preferably Windows-based)
  + Separate note files which lists instructions for TA to run software titled, “Run Instructions”
  + Any design documents documenting your decisions should also be submitted
  + Submit documents on Avenue
  + Submit hard copies of design documents in lecture

Questions:

* Could we get access to rubric? –probably by weekend, i.e. 11/5/2016
* Will we lose marks if we implement functionality that is only viable for mouse and not touch screen. i.e. hover over – potentially, wouldn’t include this
* Will we receive higher grades for hood ornaments? i.e. animations, pretty buttons, etc. – no, principle focused
* The parking meter is probably easier, how will the difficulty of marking differ between the two options? – no difference in marking
* Does innovation help a lot in higher marks? – rubric, depends on innovation
* Will we be marked poorly if area outside of touchscreen is designed poorly?
* Parking Meter:
  + Do we need to ensure it is consistent with existing designs? Lack of consistency was mentioned. – forgot to ask (I’m assuming not)
  + Do we need to design the printed ticket or use in any way? – Appears to be, can show outside area, as long as it is clearly indicated that it is separate
  + Do we need to show physical surround device, i.e. where ticket is dispensed, coins inserted? Can we? – can show outside area
  + Will we be marked on outside area?
  + How should we detect user debit/credit/coins/ticket inserted? – outside area
  + Can still enter number even if swipe, tap or insert doesn’t CVV etc. This could be provided by interface.
  + What forms of card detection should be offered? Tap/insert/swipe? Do we need to show this? – can offer swipe/insert/tap, show in outside area
  + For time – can we assume every 1 second = 1 minute? Should we list this in instructions to TA? – don’t simulate time
  + Add time to expired ticket
  + Can we assume a single user?
  + Do we need to handle a situation where there isn’t enough funds on card? – shouldn’t have a situation where not enough funds on card, handle denial
  + Do we need to handle refunding of coins if user doesn’t have enough money? - yes
  + Can it be assumed that to reset everything the TA will restart the executable/file? – shouldn’t matter
  + What about parking meter on phone? – no
  + Can we assume no debit, only credit or coins?
  + **For payment refund, can we return in coins? Do we have to keep track of how it was paid for?**
* Bank Machine:
  + For card insert – can we offer tap/swipe/insert? – didn’t get this clarified for bank machine
  + Do we need to create a system outside of the touchscreen? This could include a place to insert cards or retrieve cash or receipt. – can have outside area
  + Timeout if user has not interacted with recently? Any specifics as far as time? – Can add this functionality if wanted
  + Can we assume a single user? - no
  + Can we assume the system only needs to have a single card? i.e. will user start with the same amount of money in balance every time the system reset? – 3 sample cards should be sufficient
  + Can it be assumed that to reset everything the TA will restart the executable/file?
  + Will we have to handle multiple accounts (i.e. transfers between accounts) – yes multiple accounts, should have 3 sample cards
  + What about banking machine on phone?
* Programming Language/Coding Guidelines:
  + What specifics do we want in the instructions? How much information is enough to give the TA? – just to explain
  + What are the restrictions on file types? Can the deliverable be a .exe file? .html file? –html should be fine, avenue could take an .exe if included in a .zip?
  + Does submission file have to be an executable? – could be html
  + Will we lose grades if we accidentally discuss how to use the system in provided instructions?
  + Is a .txt or word doc sufficient for instructions?
* Submission Notes:
  + What is required for design document? – need to get this clarified
    - Include mockups?
    - How will we be marked on this? - rubric
  + How should executable and code files be submitted? Is avenue strict on this?
  + SIGCHI format?